



⑮ **BUNDESREPUBLIK  
DEUTSCHLAND**



**DEUTSCHES  
PATENT- UND  
MARKENAMT**

⑫ **Offenlegungsschrift**  
⑩ **DE 100 52 877 A 1**

⑤ Int. Cl.<sup>7</sup>:  
**G 06 F 9/44**  
G 05 B 19/042

⑲ Aktenzeichen: 100 52 877.5  
⑳ Anmeldetag: 19. 10. 2000  
㉓ Offenlegungstag: 2. 8. 2001

**DE 100 52 877 A 1**

③① Unionspriorität:

99-45785 21. 10. 1999 KR  
2000-61457 19. 10. 2000 KR

⑦① Anmelder:

Samsung Electronics Co., Ltd., Suwon, Kyonggi, KR

⑦④ Vertreter:

Patentanwälte Ruff, Wilhelm, Beier, Dauster &  
Partner, 70174 Stuttgart

⑦② Erfinder:

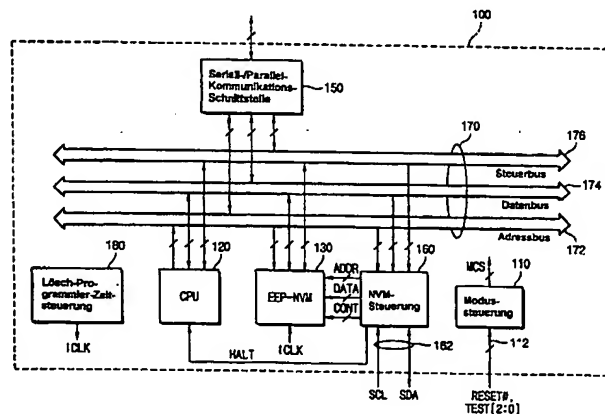
Ahn, Jong-Keun, Seoul/Soul, KR; Yu, Bum-Seok,  
Seoul/Soul, KR; Ra, Sang-Joo, Seoul/Soul, KR; Kim,  
Jong-In, Seoul/Soul, KR

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤④ Mikrocontroller

⑤⑦ Die Erfindung bezieht sich auf einen Mikrocontroller mit einer mit mehreren Bussen (170) verbundenen Zentralprozessoreinheit (CPU), einer Modussteuereinheit (110) zur Erzeugung von Modussteuersignalen für die Einstellung verschiedener Betriebsarten, einer Kommunikationsschnittstelle und einem elektrisch wiederbeschreibbaren, nichtflüchtigen Speicher (130). Erfindungsgemäß wird durch geeignete Auslegung der Speichersteuereinheit bewirkt, dass beim Schreiben eines Steuerprogramms für den Mikrocomputer in den nichtflüchtigen Speicher abhängig von einem Programmierbefehl die CPU die Steuerung des nichtflüchtigen Speichers nicht beeinflusst. Insbesondere können in einem Programmiermodus die CPU, die Busse, I/O-Anschlüsse und die Kommunikationsschnittstelle des Mikrocontrollers deaktiviert werden. Die Speichersteuereinheit empfängt einen Schreibbefehl und eine Adresse von außerhalb des Mikrocontrollers und steuert den Betrieb des nichtflüchtigen Speichers basierend auf dem Schreibbefehl und der Adresse ohne Eingriff der CPU. Verwendung in der Mikrocontrollertechnologie.



**DE 100 52 877 A 1**

## Beschreibung

Die Erfindung bezieht sich auf einen Mikrocontroller nach dem Oberbegriff des Anspruchs 1.

Die Verwendung von VLSI-Chipschaltkreisen in digitalen Datenverarbeitungssystemen bzw. Computern hat zu Vorteilen hinsichtlich kleinerer Abmessung, niedrigerem Gewicht, geringeren Kosten, reduziertem Leistungsverbrauch und höherer Zuverlässigkeit geführt. Entsprechende VLSI-Schaltkreise, wie Einzelchip-Computer, d. h. Mikrocomputer oder Mikrocontroller, erlauben die Nutzung von Computerelektronik in praktisch allen heutigen Anwendungen sowohl zu Hause als auch kommerziell in der Industrie.

Ein Steuer- oder Anwendungssoftware-Programm zum Betrieb eines Mikrocontrollers ist typischerweise in einem Festwertspeicher (ROM), wie einem maskenprogrammierten Festwertspeicher, kurz Masken-ROM, einem löschbaren und programmierbaren Festwertspeicher (EPROM), einem elektrisch löschbaren und programmierbaren Festwertspeicher (EEPROM) oder einem Flash-EEPROM gespeichert. Es ist allgemein bekannt, dass es nicht möglich ist, den Inhalt eines eingebauten maskenprogrammierten Festwertspeichers zu ändern, sobald dieser gespeichert ist. Ein in einem Masken-ROM gespeichertes Programm kann daher nicht erneuert werden. Im Gegensatz dazu kann ein Mikrocontroller mit einem eingebauten EPROM, EEPROM oder Flash-EEPROM insgesamt neu beschrieben oder teilweise aktualisiert werden, wenn ein Steuerprogramm desselben zu ändern ist.

Gegenwärtig finden Mikrocontroller mit eingebauten Flash-Speichern in einer Vielzahl von Anwendungen breiten Gebrauch, wie in Mobiltelefonen, Anrufer-ID-Boxen, CD-ROM-Treibern, DVD-Spielern, Smartcards und Settop-Boxen. Solche Mikrocontroller mit eingebautem Flash-Speicher können programmiert werden, während sie auf einer gedruckten Leiterplatte eines Systems montiert sind, d. h. in einem "Onboard"-Zustand.

Ein typischer Mikrocontroller kann eine Zentralprozessoreinheit (CPU), ein eingebautes Flash-EEPROM zur Speicherung eines Steuerprogramms für die CPU, ein ROM zur Speicherung eines Schreibprogramms für das Schreiben des Steuerprogramms in den Flash-Speicher, einen Speicher mit wahlfreiem Zugriff (RAM) zur Speicherung temporärer Daten und eine Modussteuereinheit zum Einstellen verschiedener Betriebsarten des Mikrocontrollers enthalten.

Bei einem z. B. in der Patentschrift US 5.872.994 beschriebenen Mikrocontroller mit eingebautem Flash-Speicher wird in einer Onboard-Schreibbetriebsart eine Steuerprogrammquelle über einen Adapter mit einer seriellen Kommunikationsschnittstelle oder einem Eingabe/Ausgabe-Anschluss verbunden. Wenn eine Modussteuereinheit durch ein vom Eingabe/Ausgabe-Anschluss oder dergleichen zugeführtes Datensignal auf einem Datenbus die Onboard-Schreibbetriebsart erkennt, verbindet sie die serielle Kommunikationsschnittstelle, einen Flash-Speicher, ein RAM und eine CPU mit Datenbussen. In dieser Betriebsart liest die CPU das Steuerprogramm über die serielle Kommunikationsschnittstelle aus der Steuerprogrammquelle gemäß dem im ROM gespeicherten Schreibprogramm und speichert temporär das Steuerprogramm in das RAM.

Das ROM des bekannten Mikrocontrollers ist jedoch nur in einem Onboard-Schreibmodus aktiv, während es in anderen Betriebsarten inaktiv ist. Dementsprechend limitiert der Einbau eines solchen ROMs in einen Mikrocontrollerchip den Vorteil der Chipabmessungsreduktion des Mikrocontrollers. Es ist auch eine alternative Onboard-Schreibmethode bekannt, bei der ein RAM anstelle eines ROM verwendet wird, um ein Schreibprogramm zu speichern. Diese

Methode reduziert die Abmessung des Mikrocontrollers, da das ROM zur Speicherung eines Schreibprogramms nicht mehr benötigt wird. Dennoch beinhaltet diese Technologie noch ein RAM als wesentliches Element.

Des weiteren beinhaltet der bekannte Mikrocontroller einen Datenempfangszeitdetektor, der die Lesegeschwindigkeit eines von einer Programmquelle zugeführten Programms durch Überwachen der Systemtaktgeschwindigkeit während eines Onboard-Schreibvorgangs detektiert. Bei hoher Lesegeschwindigkeit speichert die CPU temporär das Programm in das RAM oder einen Puffer der CPU, bevor sie das Programm in den Flash-Speicher schreibt. Bei niedriger Lesegeschwindigkeit schreibt die CPU hingegen das Programm direkt in den Flash-Speicher. Außerdem kann der Mikrocontroller nach dem Stand der Technik die Zeitperiode für einen Schreibvorgang oder einen Flash-Löschvorgang variieren. Diese Merkmale neigen jedoch dazu, den Aufwand und die Schaltkreiskomplexität des Mikrocontrollers zu erhöhen.

Weitere Beispiele bekannter Mikrocomputer mit eingebautem Flash-Speicher sind in den Patentschriften US 5.398.208 und US 5.493.534 beschrieben, auf die insoweit verwiesen werden kann.

Der Erfindung liegt als technisches Problem die Bereitstellung eines Mikrocontrollers der eingangs genannten Art zugrunde, mit dem sich eine vergleichsweise geringe Chipabmessung bei relativ geringem Aufwand erreichen lässt.

Die Erfindung löst dieses Problem durch die Bereitstellung eines Mikrocontrollers mit den Merkmalen des Anspruchs 1. Der erfindungsgemäße Mikrocontroller ist in der Lage, Onboard-Programmiervorgänge unter Verwendung eines elektrisch wiederbeschreibbaren nichtflüchtigen Speichers durchzuführen, ohne zusätzliche Speicher zum Speichern eines Steuerprogrammsatzes und entsprechender Daten zu benötigen. Der Mikrocontroller lässt sich unabhängig von der Systemumgebung auf eine konstante Löschozeit und Programmierzeit auslegen, so dass Probleme hinsichtlich zu weitgehender Löschung und Schreibstörung vermieden werden. Der Mikrocontroller benötigt relativ wenig Chipfläche, was die Herstellungskosten senkt. Die erfindungsgemäße Onboard-Programmierarchitektur ist für alle Mikrocontroller mit beliebigen Typen elektrisch programmierbarer, nichtflüchtiger Halbleiterspeicher unabhängig von deren Leistungsvermögen anwendbar, d. h. sie ist sowohl für Mikrocomputer der höheren wie auch der niedrigeren Preisklasse einsetzbar.

Der erfindungsgemäße Mikrocontroller beinhaltet eine Speichersteuereinheit für den elektrisch wiederbeschreibbaren nichtflüchtigen Speicher und eine Modussteuereinheit zum Einstellen verschiedener Betriebsarten, wie eines "Tool"- oder "Werkzeug"-Modus und eines "Nutzerprogrammier"-Modus. Im Werkzeugmodus werden die Zentralprozessoreinheit und eine oder mehrere vorhandene Kommunikationsschnittstellen deaktiviert. Die Speichersteuereinheit steuert dann einen internen Vorgang, wie einen Lesen-, Schreib- oder Löschvorgang, für den elektrisch wiederbeschreibbaren nichtflüchtigen Speicher ohne Eingriff der Zentralprozessoreinheit basierend auf einem seriell zugeführten Satz aus einem Befehl, einer Adresse und Daten. Im Nutzerprogrammiermodus empfängt die Zentralprozessoreinheit einen Befehl bezüglich eines internen Vorgangs sowie dazugehörig eine Adresse und Daten und stellt die Speichersteuereinheit abhängig von dem Befehl ein. Im Fall eines Lesen- oder Löschvorgangs leitet die Zentralprozessoreinheit die Adresse an die Speichersteuereinheit weiter, im Fall eines Schreibvorgangs auch die entsprechenden Daten. Während eines Schreibvorgangs hält die Speichersteuereinheit dann die Zentralprozessoreinheit an.

Vorteilhafte Weiterbildungen der Erfindung sind in den Unteransprüchen angegeben.

Vorteilhafte Ausführungsformen der Erfindung sind in den Zeichnungen dargestellt und werden nachfolgend beschrieben. Hierbei zeigen:

Fig. 1 ein Blockschaltbild eines ersten erfindungsgemäßen Mikrocontrollers,

Fig. 2 ein Blockschaltbild wesentlicher Bestandteile des Mikrocontrollers von Fig. 1,

Fig. 3 ein Zeitsteuerungsdiagramm eines Nutzerprogrammiermodus des Mikrocontrollers von Fig. 1,

Fig. 4 bis 7 Darstellungen zur Illustration von Schlüsselregistern, eines Nutzerprogrammiersteuerregisters, eines Nutzerprogrammierfehlerregisters bzw. eines Sektorschutzhinweisregisters innerhalb einer Speichersteuereinheit von Fig. 1 und

Fig. 8 ein Blockschaltbild eines zweiten erfindungsgemäßen Mikrocontrollers.

Im folgenden werden vorteilhafte Ausführungsbeispiele der Erfindung unter Bezugnahme auf die Zeichnungen näher erläutert, wobei im wesentlichen nur die jeweils erfindungsrelevanten Komponenten gezeigt sind, während der Übersichtlichkeit halber allgemein bekannte Komponenten, wie Schaltkreisblöcke, elementare Strukturen und architekturbezogene Funktionalitäten, nicht näher gezeigt und beschrieben werden.

Fig. 1 veranschaulicht einen ersten erfindungsgemäßen Mikrocontroller 100 in Form einer Einzelchip-Datenverarbeitungsvorrichtung mit z. B. 32 Bit. Der Mikrocontroller 100 beinhaltet eine Modussteuereinheit 110, eine Zentralprozessoreinheit (CPU) 120, einen elektrisch löschbaren und programmierbaren, nichtflüchtigen Halbleiterspeicher (EEP-NVM) 130, wie ein EEPROM oder ein Flash-EEPROM, der als Programmspeicher des Mikrocontrollers 100 verwendet wird. Die CPU 120 und der EEP-NVM 130 sind mit Bussen 170 verbunden, die einen Adressbus 172, einen Datenbus 174 mit z. B. 32 Bit sowie einen Steuerbus 176 umfassen.

Die Modussteuereinheit 110 empfängt Modusauswahlsignale RESET#, TEST0, TEST1 und TEST2, die über einen Eingangsanschluß 112 dem Mikrocontroller 100 von außen zugeführt werden, und erzeugt Modussteuersignale MCS zum Einstellen verschiedener Betriebsarten in jedem Abschnitt des Mikrocontrollers 100. Bevorzugt werden die MCS von einer Logikkombination der Modusauswahlsignale RESET# und TEST0 bis TEST2 abgeleitet. Mögliche Betriebsarten des Mikrocomputers 100 umfassen einen "Tool"- oder "Werkzeug"-Modus als einen ersten Programmiermodus, einen "Nutzerprogrammier"-Modus als einen zweiten Programmiermodus, mehrere Testmodi etc.

Der Mikrocontroller 100 beinhaltet des weiteren eine oder mehrere bitprogrammierbare Kommunikationsschnittstellen 150, die wenigstens eine serielle Schnittstelle und/oder wenigstens eine parallele Schnittstelle umfassen. Beispielhafte Kommunikationsschnittstellen, die erfindungsgemäß verwendbar sind, umfassen solche vom Typ synchrone SIO, UART oder I<sup>2</sup>C. Die Kommunikationsschnittstelle 150 ist selektiv in Abhängigkeit von den Modussteuersignalen MCS der Modussteuereinheit 110, der internen Moduseinstellsignale oder des Moduseinstellbefehls mit den internen Bussen 170 gekoppelt und kann bei Bedarf Eingabe/Ausgabe(I/O)-Anschlüsse beinhalten, die bitprogrammierbar sein können. Alternativ können für den Mikrocontroller 100 zusätzlich zu Kommunikationsschnittstellen I/O-Anschlüsse verwendet werden, um externe Kommunikationsgeräte anzubinden. Die CPU 120 steuert den EEP-NVM 130, die I/O-Anschlüsse 140 und die Kommunikationsschnittstellen 150 in Abhängigkeit von den Betriebsarten des Mikrocontrollers

100.

Der Mikrocontroller 100 weist zusätzlich eine NVM-Steuereinheit 160 mit einer seriellen Schnittstelle, siehe Bezugszeichen 164 von Fig. 2, und einer parallelen Schnittstelle, siehe Bezugszeichen 166 von Fig. 2, auf. Die serielle Schnittstelle der NVM-Steuereinheit 160 weist einen seriellen Anschluss 162 mit einem Takteingabeanschluss SCL und einem I/O-Anschluss SDA für serielle Daten auf. Dabei kann der serielle Anschluss 162 in anderen Ausführungsformen der Erfindung zusätzlich einige weitere Anschlüsse umfassen, wie einen Anschluss für eine Spannungsversorgung V<sub>DD</sub>, einen Anschluss für Masse V<sub>SS</sub>, Testanschlüsse und einige reservierte Anschlüsse.

Die NVM-Steuereinheit 160 empfängt einen Befehl und eine Adresse von außerhalb des Mikrocomputers 100 über den I/O-Datenanschluss SDA, die I/O-Anschlüsse 140 oder die Kommunikationsschnittstelle 150 und steuert direkt die Betriebsvorgänge des EEP-NVM 130 auf der Basis des Befehls und der Adresse ohne Einwirkung der CPU 120.

Im Werkzeugmodus, in welchem der Mikrocontroller 100 mit einem Satz von Steuerprogrammen zur Steuerung des Mikrocontrollers 100 programmiert wird, empfängt die NVM-Steuereinheit 160 über den I/O-Datenanschluss SDA einen Satz aus einem Modusbefehl, Daten und einer Adresse von außerhalb des Mikrocontrollers 100 und steuert direkt den Betrieb des EEP-NVM 130 auf der Basis des Befehls und der Adresse ohne Einwirkung der CPU 120.

In einem derartigen seriellen Programmiermodus des Mikrocontrollers 100 ordnet die NVM-Steuereinheit 160 die seriellen Daten einschließlich eines Befehlendes von z. B. 4 Bit, eines Adressfeldes von z. B. 20 Bit und eines Datenfeldes in einem parallelen Datenformat an und führt dem EEP-NVM 130 parallel eine Adresse ADDR und Daten DATA zusammen mit einigen Steuersignalen CONT zu. Anschließend steuert die NVM-Steuereinheit 160 einen Schreibvorgang des EEP-NVM 130 basierend auf dem Schreibbefehl und der Adresse. Nach Abschluß des obigen Programmiervorgangs kann der EEP-NVM 130 als Boot-Speicher des Mikrocontrollers 100 verwendet werden.

Wie oben angegeben, ist der erfindungsgemäße Mikrocontroller in der Lage, einen Onboard-Programmiervorgang ohne irgendwelche anderen Speicher außer dem EEP-NVM 130 durchzuführen, wie ein ROM und/oder ein RAM, wodurch eine geringere Chipabmessung und niedrigere Kosten erzielt werden.

Andere Betriebsarten, wie beispielsweise der zu einem Schreibmodus des EEP-NVM 130 gehörigen Nutzerprogrammiermodus, Mikrocontroller-Testmodi und andere Betriebsarten des EEP-NVM, d. h. zugehörige Löscho- und Lesemodi, werden durch einen Moduseinstellbefehl gesetzt, der von einem externen Steuerprogrammquellengerät abgegeben wird, wie einer ROM-Schreibeinheit oder einem Hauptcomputer. In diesem zweiten Programmiermodus deaktiviert die Modussteuereinheit 110 den seriellen I/O-Anschluß 162, der mit der NVM-Steuereinheit 160 verbunden ist, und die CPU 120 empfängt über die Kommunikationsschnittstelle 150 einen Befehl für einen internen Vorgang, um diesen Befehl zu decodieren. Anschließend setzt die CPU 120 in der NVM-Steuereinheit 160 zugehörige Steuerbits in Reaktion auf den decodierten Befehl für einen internen Vorgang. Wenn der decodierte Befehl einen Lese- oder Löscho-Vorgang anzeigt, entnimmt die CPU 120 die betreffende Adresse von der Kommunikationsschnittstelle 150 und sendet sie zur NVM-Steuereinheit 160. Wenn der decodierte Befehl einen Schreibvorgang anzeigt, sendet die CPU 120 die betreffende Adresse und die zugehörigen Daten von der Kommunikationsschnittstelle 150 zur NVM-Steuereinheit 160. Danach hält die NVM-Steuereinheit 160 die CPU

120 an, bis der Schreibvorgang für den EEP-NVM 130 abgeschlossen ist.

Wenn ein EEPROM oder ein Flash-EEPROM als der EEP-NVM 130 verwendet wird, beinhaltet der Mikrocontroller 100 vorzugsweise eine Steuereinheit 180 zur Löscho- und Programmierzeitsteuerung. Diese Steuereinheit 180 erlaubt ein Löschen oder Programmieren des EEPROM oder des Flash-Speichers mit einer konstanten Löscho- oder Programmiergeschwindigkeit unabhängig von der Umgebung des Systems, für das der erfindungsgemäße Mikrocontroller 100 eingesetzt wird. Die Steuereinheit 180 zur Programmier- und Löschozeitsteuerung beinhaltet einen nicht gezeigten Taktgenerator zur Erzeugung eines Löscho- und/oder Programmiersynchronisiertaktsignals ICLK konstanter Frequenz, das dazu dient, den EEPROM oder Flash-Speicher zu takten. Dadurch kann das EEPROM oder der Flash-Speicher mit konstanter Geschwindigkeit unabhängig von einem Systemhaupttaktsignal gelöscht oder programmiert werden, das an den Mikrocontroller 100 von außen angelegt wird, was Probleme bezüglich übermäßigem Löschen und Schreibstörung für das EEPROM oder den Flash-Speicher vermeidet.

Fig. 2 zeigt als Blockdiagramm exemplarische Verbindungen der CPU 120, des EEP-NVM 130 und der NVM-Steuereinheit 160. Wie daraus ersichtlich, beinhaltet der EEP-NVM 130 ein nicht-flüchtiges Speicherzellenfeld 132, wie ein EEPROM-Zellenfeld, ein Flash-Speicherzellenfeld oder ein ferroelektrisches Speicherzellenfeld. Des weiteren weist der EEP-NVM 130 einen Datenpuffer 134 zum temporären Speichern von Daten während einer langen Schreibdauer für z. B. EEPROMs oder Flash-Speicher auf. Zum selben Zweck kann die CPU 120 auch eine Registerdatei 122 aufweisen.

Im Fall der Verwendung eines EEPROM oder Flash-Speichers für den EEP-NVM 130 beinhaltet dieser eine Steuerungschaltung 136 für hohe Spannung, wie eine Ladungspumpschaltung, zur Durchführung seiner Programmier- und Löschovorgänge. Des weiteren ist ein jeweiliges Speicherzellenfeld des EEPROMs oder Flash-Speichers in mehrere Sektoren SEC1, SEC2, . . . , SECn unterteilt und mit einer Löscho- und/oder Schreibsperrfunktion für jeden einzelnen Sektor ausgerüstet, so dass das EEPROM oder der Flash-Speicher löscho- und/oder programmiertespernte Sektoren erkennen und ein fehlerhaftes Löschen und/oder Beschreiben von löscho- und/oder schreibgesperrten Zellensektoren verhindern kann. Diese Funktionen werden weiter unten näher erläutert.

Die NVM-Steuereinheit 160 weist eine serielle Schnittstelle 164 und eine Speicherschnittstellenschaltung 166 auf. Die serielle Schnittstelle 164 dient zur Anbindung von Komponenten außerhalb des Mikrocontrollers 100 im Werkzeugmodus. Die Speicherschnittstellenschaltung 166 weist ein Adressenregister 166a, ein Datenregister 166b, eine Steuerregisterschaltung 166c und einen Adressenselektor 166d auf.

Im Werkzeugmodus werden Adressen, Daten und Befehle von der seriellen Schnittstelle 164 dem Adressenregister 166a, dem Datenregister 166b und der Steuerregisterschaltung 166c zugeführt. Außerdem sind das Adressenregister 166a an den Adressbus 172, das Datenregister 166b an den Datenbus 174 und die Steuerregisterschaltung 166c an den Steuerbus 176 angeschlossen. Im Nutzerprogrammiermodus empfängt jedoch die CPU 120 einen Befehl für einen internen Vorgang über die Kommunikationsschnittstelle 150, um diesen Befehl zu decodieren. Anschließend setzt die CPU 120 in der Steuerregisterschaltung 166c Steuerbits fest. Wenn der decodierte Befehl einen Lese- oder Löschovorgang anzeigt, holt sich die CPU 120 Adressen von der

Kommunikationsschnittstelle 150 und legt sie im Adressenregister 166a ab. Wenn der decodierte Befehl einen Schreibvorgang anzeigt, holt sich die CPU 120 Adressen und Daten von der Kommunikationsschnittstelle 150 und legt die Adressen im Adressenregister 166a und die Daten im Datenregister 166b über den Adressbus 172 bzw. den Datenbus 174 ab. Anschließend hält die NVM-Steuereinheit 160 die CPU 120 an, bis der Schreibvorgang für den EEP-NVM 130 abgeschlossen ist.

Die Betriebsarten des Mikrocontrollers 100 einschließlich des Werkzeugmodus, des Nutzerprogrammiermodus und einiger Testmodi werden durch eine jeweilige Kombination der Modusauswahlbefehlssignale RESET# und REST0 bis TEST2 festgesetzt, die dem Eingangsanschluß 112 der Modussteuereinheit 110 zugeführt werden.

Im Werkzeugmodus deaktiviert die Modussteuereinheit 110 die CPU 120 und die Kommunikationsschnittstelle 150, bis ein anderer Betriebsmodus aktiviert wird. Im Werkzeugmodus werden der Befehl, die Adresse und die Daten vorzugsweise seriell der NVM-Steuereinheit 160 über den seriellen I/O-Anschluss 162 zugeführt. Dadurch ist der Mikrocontroller 100 in der Lage, eine Onboard-Programmierung des EEP-NVM 130 ohne einen weiteren Schreibprogrammspeicher, wie ein ROM oder ein RAM, auszuführen. Die NVM-Steuereinheit 160 schreibt direkt einen Steuerprogrammsatz, d. h. entsprechende Daten, in den adressierten Bereich des EEP-NVM 130 gemäß der empfangenen Adresse. Der so programmierte EEP-NVM 130 kann als Boot-Speicher des Mikrocontrollers 100 verwendet werden. Es versteht sich, dass ein solcher Steuerprogrammsatz auch in einem Offboard-Zustand des Mikrocontrollers 100 in den EEP-NVM 130 unter Verwendung eines ROM-Schreibgerätes, eines Hauptcomputers oder dergleichen geschrieben werden kann.

Im Nutzerprogrammiermodus deaktiviert die Modussteuereinheit 110 den seriellen I/O-Anschluss 162. Die CPU 120 liest einen Moduseinstellbefehl von der Kommunikationsschnittstelle 150 und setzt vorzugsweise fünf Stellen der Internbetrieb-Steuerbits der Steuerregisterschaltung 166c mittels Decodieren des Befehls. Wenn der decodierte Befehl für einen internen Vorgang einen Lese- oder Löschovorgang anzeigt, bewegt die CPU 120 eine Adresse von der Kommunikationsschnittstelle 150 zum Adressenregister 166a. Wenn der decodierte Befehl einen Schreibvorgang anzeigt, bewegt die CPU 120 eine Adresse und Daten von der Kommunikationsschnittstelle 150 zum Adressenregister 166a bzw. zum Datenregister 166b. Danach hält die NVM-Steuereinheit 160 die CPU 120 an, bis der Schreibvorgang für den EEP-NVM 130 abgeschlossen ist.

Im Nutzerprogrammiermodus gibt die CPU 120 einen Schreibbefehl ab, welcher der NVM-Steuereinheit 160 anzeigt, dass der EEP-NVM 130 in einen Schreibvorgang für ein Aktualisierungsprogramm eintritt, das zu einem Teil des im EEP-NVM 130 gespeicherten Steuerprogrammsatzes gehört. Der Befehl wird über die NVM-Steuereinheit 160 an den EEP-NVM 130 übertragen. Das Aktualisierungsprogramm, d. h. ein Teil des Steuerprogrammsatzes, wird der NVM-Steuereinheit 160 über die Busse 170 von einem Gerät außerhalb des Mikrocontrollers 100 zugeführt und zur Aktualisierung des im EEP-NVM 130 gespeicherten Steuerprogrammsatzes verwendet. Das Aktualisierungsprogramm wird ebenfalls in den EEP-NVM 130 geschrieben, während der Mikrocontroller 100 auf einer Leiterplatte eines Systems montiert ist. Andererseits hält die NVM-Steuereinheit 160 die CPU 120 dadurch an, dass sie ein CPU-Haltesignal HALT in Reaktion auf den Schreibbefehl von der CPU 120 erzeugt. Wie in Fig. 3 dargestellt, die ein Zeitsteuerungsdiagramm des Nutzerprogrammiermodus zeigt, hält ein CPU-

Wartesignal die Verarbeitung durch die CPU 120 an, während das Aktualisierungsprogramm in den EEP-NVM 130 geschrieben wird. Während dieser CPU-Wartezeit werden sowohl ein Freigabesignal HVE für hohe Spannung als auch ein Programmier-/Löschsteuersignal Programmieren/Löschen für EEPROMs und Flash-Speicher aktiv.

Fig. 4 veranschaulicht Schlüsselregister, die in der NVM-Steuereinheit 160 enthalten sind. Gemäß Fig. 4 sind vier Register KR1-KR4 mit je acht Bit vorgesehen. Die Schlüsselregister KR1-KR4 verhindern, dass innerhalb der NVM-Steuereinheit 160 auf die Steuerregister zugegriffen wird, solange nicht erste bis vierte vorgegebene Schlüsselwerte in das jeweilige Register KR1-KR4 geschrieben werden. Speziell können die Register in der Steuerregisterschaltung 166c nur dann beschrieben werden, wenn die Register KR1-KR4 die ersten bis vierten Schlüsselwerte enthalten, z. B. die Werte "01011010" (= 5A hexadezimal), "10100101" (= A5 hexadezimal), "01011010" bzw. "10100101".

Fig. 5 veranschaulicht ein Format eines in der NVM-Steuereinheit 160 enthaltenen, bitprogrammierbaren Nutzerprogrammiersteuerregisters mit 16 Bit. Gemäß Fig. 5 beinhaltet das bitprogrammierbare Register ein Hochgeschwindigkeitsbit FAST, Zugriffzyklusbits ACC0-ACC2 zum Zuweisen von Zugriffzyklen für den EEP-NVM, ein Softwaredatenschutz-Sperribit SDPD, ein Bit STRSTP zum Starten und Stoppen des Betriebs, ein Sektorprogrammierfreigabebit SPGM, ein Optionssektorprogrammierfreigabebit OSPGM, ein Sektorlöschfreigabebit SERS insbesondere für EEPROMs und ein Optionssektorlöschfreigabebit OSERS insbesondere für EEPROMs. Alle sechzehn Bit des bitprogrammierbaren Registers sind anfänglich auf logisch "0" gesetzt.

Wenn das Hochgeschwindigkeitsbit FAST auf logisch "1" gesetzt wird, ist der interne Schaltungsaufbau des EEP-NVM 130 für einen Betriebsmodus mit hoher Geschwindigkeit konfiguriert, z. B. für einen Verschachtelungsmodus. Die Zugriffzyklusbits ACC0 bis ACC2 bezeichnen Zugriffzyklen für den EEP-NVM. Beispielsweise bezeichnen "000" 3 Zyklen, "001" 1 Zyklus, "010" 2 Zyklen und "011" den Zustand "Bank gesperrt".

Das Konfigurieren der Nutzerprogrammiersteuerregister beginnt mit dem Setzen des Softwaredatenschutz-Sperribits SDPD und eines der Bits SPGM, OSPGM, SERS und OSERS auf logisch "1". In diesem Zustand beginnt der EEP-NVM 130 mit dem Betrieb, wenn das Bit STRSTP auf logisch "1" gesetzt wird. Während des Betriebs des EEP-NVM 130 hält die CPU 120 an und verbleibt in einem Wartezustand, wie oben beschrieben. Die CPU 120 arbeitet nicht vor Abschluss des Betriebs des EEP-NVM 130.

Fig. 6 veranschaulicht ein Format eines in der NVM-Steuereinheit 160 enthaltenen Nutzerprogrammierfehlerregisters. Wie aus Fig. 6 ersichtlich, enthält das Fehlerregister ein Konfigurationsfehlerbit CFGERR, ein Programmierfehlerbit PGMERR und ein Spannungsfehlerbit VTGERR.

Das Konfigurationsfehlerbit CFGERR ist auf logisch "1" eingestellt, wenn mehr als zwei der Bits SPGM, OSPGM, SERS und OSERS des Nutzerprogrammiersteuerregisters von Fig. 5 gleichzeitig auf logisch "1" gesetzt sind. Sobald das Konfigurationsfehlerbit CFGERR auf logisch "1" gesetzt wurde, kann auf das Nutzerprogrammiersteuerregister nicht mehr zugegriffen werden, bis das Bit CFGERR auf logisch "0" zurückgesetzt wurde.

Das Programmierfehlerbit PGMERR wird auf logisch "1" gesetzt, wenn ein Programmierdatenwert nicht mit einem programmierten Datenwert im Programmierbetrieb übereinstimmt. Der programmierte Datenwert wird intern direkt nach dem Programmiervorgang mit dem programmierten

Datenwert verglichen. Das Spannungsfehlerbit VTGERR ist auf logisch "1" gesetzt, wenn Betriebsspannungen, wie Programmierspannungen und Löschspannungen eines EEPROM oder Flash-Speichers, unter untere Grenzwerte abfallen oder über obere Grenzwerte ansteigen.

Fig. 7 veranschaulicht ein Format eines in der NVM-Steuereinheit 160 enthaltenen Sektorschutzhinweisregisters für den Fall, dass der EEP-NVM 130 von einem EEPROM oder Flash-Speicher mit 256 Kbyte gebildet ist, dessen Speicherzellenfeld in sechzehn Sektoren mit je 2 Kbyte und sieben Sektoren mit je 32 Kbyte aufgeteilt ist. Wie aus Fig. 7 ersichtlich, bilden Bits L0 bis LF Schutzmarkierungen für die sechzehn Sektoren mit je 2 Kbyte, und Bits U1 bis U7 bilden Schutzmarkierungen für die sieben Sektoren mit je 32 Kbyte. Wenn ein jeweiliges Hinweisbit auf logisch "1" gesetzt wird, wird der zugehörige Sektor bzgl. Löschen/Programmieren gesperrt. Es sei angemerkt, dass das Sektorschutzhinweisregister statt in der NVM-Steuereinheit im Zellenfeld des EEP-NVM oder dessen Peripherie implementiert sein kann.

Fig. 8 veranschaulicht einen zweiten erfindungsgemäßen Mikrocontroller 100', der in seinem Aufbau im wesentlichen demjenigen von Fig. 1 mit der Ausnahme entspricht, dass er zusätzlich ein Puffer-RAM 190, wie ein statisches RAM (SRAM) oder ein ferroelektrisches RAM (FRAM), beinhaltet und I/O-Anschlüsse 140 statt oder zusätzlich zur Kommunikationsschnittstelle 150 vorgesehen sind. Der Pufferspeicher 190 ist mit den Bussen 170 verbunden und wird zur temporären Speicherung von Daten, wie einem Aktualisierungsprogramm, z. B. während des Nutzerprogrammiermodus verwendet, in dem das Aktualisierungsprogramm, d. h. heißt ein Teil des Steuerprogramms, in den EEP-NVM 130 geschrieben wird. In diesem Fall stellt die CPU 120 die Steuerregisterschaltung 166c der NVM-Steuereinheit 160 (siehe Fig. 2) so ein, dass die im Pufferspeicher 190 gespeicherten Daten über die NVM-Steuereinheit 160 in den EEP-NVM 130 geschrieben werden. Außerdem wird der Pufferspeicher 190 durch die Modussteuereinheit 110 im Werkzeugmodus, in welchem der Mikrocontroller mit einem kompletten Satz von Steuerprogrammen programmiert wird, zusammen mit der CPU 120, den I/O-Anschlüssen 140, den Kommunikationsschnittstellen 150 und den Bussen 170 deaktiviert.

Alternativ werden externe Daten temporär in einer Registerdatei 122 innerhalb der CPU 120 gespeichert. In diesem Fall stellt die CPU 120 die Steuerregisterschaltung 166c der NVM-Steuereinheit 160 so ein, dass die in der Registerdatei 122 gespeicherten Daten z. B. während des Nutzerprogrammiermodus über die NVM-Steuereinheit 160 in den EEP-NVM 130 geschrieben werden.

In einer weiteren erfindungsgemäßen Ausführungsform werden externe Daten temporär in einem Datenpuffer 134 des EEP-NVM 130, siehe Fig. 2, gespeichert. Die CPU 120 stellt dann die Steuerregisterschaltung 166c der NVM-Steuereinheit 160 so ein, dass die im Datenpuffer 134 gespeicherten Daten während Betriebsarten wie z. B. des Nutzerprogrammiermodus über die NVM-Steuereinheit 160 in den EEP-NVM 130 geschrieben werden.

Im übrigen entspricht die Funktionsweise des Mikrocontrollers von Fig. 8 derjenigen des Mikrocontrollers von Fig. 1, mit Ausnahme der oben beschriebenen Unterschiede hinsichtlich des Datenpufferbetriebs.

Die oben erläuterten Ausführungsbeispiele zeigen, dass die Erfindung einen Mikrocontroller bereitstellt, der in der Lage ist, einen Onboard-Programmierbetrieb ohne jegliche weitere Speicher außer dem EEP-NVM zum Speichern von Daten oder eines Steuerprogramms auszuführen. Außerdem kann der Mikrocontroller eine konstante Löschdauer und

eine konstante Programmierdauer unabhängig von der Systemumgebung besitzen, so dass keine Probleme hinsichtlich übermäßiger Löschung und Schreibstörung auftreten. Zudem benötigt der Mikrocontroller nur eine reduzierte Chipfläche, was die Herstellungskosten senkt. Die erfindungsgemäße Onboard-Programmierarchitektur ist für alle Mikrocontroller beliebigen Typs mit elektrisch programmierbaren, nichtflüchtigen Halbleiterspeichern unabhängig von deren Leistungsvermögen anwendbar, d. h. die erfindungsgemäße Architektur kann für Mikrocontroller mit elektrisch programmierbaren, nichtflüchtigen Halbleiterspeichern sowohl der höheren als auch der niedrigeren Preisklasse benutzt werden.

## Patentansprüche

1. Mikrocontroller mit
  - einer Zentralprozessoreinheit (120), die an eine Mehrzahl von Bussen (170) angeschlossen ist,
  - einer Modussteuereinheit (110) zur Erzeugung von Modussteuersignalen zum Einstellen von Betriebsarten des Mikrocontrollers in Reaktion auf ein Modusauswahlsignal,
  - einer mit den Bussen verbundenen Kommunikationsschnittstelle (150) zur externen Geräteanbindung und
  - einem elektrisch wiederbeschreibbaren, nichtflüchtigen Speicher (130),
 dadurch gekennzeichnet, dass
  - der elektrisch wiederbeschreibbare, nichtflüchtige Speicher (130) zur Speicherung von Daten und eines Steuerprogramms dient und
  - eine mit dem elektrisch wiederbeschreibbaren, nichtflüchtigen Speicher verbundene Speichersteuereinheit (160) vorgesehen ist, die funktionell mit den Bussen (170) und/oder der Kommunikationsschnittstelle (150) gekoppelt ist, um einen Befehl für den nichtflüchtigen Speicher, eine Adresse und/oder Daten in Abhängigkeit von den Modussteuersignalen zu empfangen,
2. Mikrocontroller nach Anspruch 1, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit (160) eine serielle Schnittstelle enthält und über die Kommunikationsschnittstelle und/oder die serielle Schnittstelle den Befehl für den nichtflüchtigen Speicher sowie die Adresse und/oder die Daten empfängt.
3. Mikrocontroller nach Anspruch 1 oder 2, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit (160) in einer der Betriebsarten direkt ohne Einwirkung der Zentralprozessoreinheit (120) die Daten und die Adresse aus dem nichtflüchtigen Speicher (130) liest und/oder in diesen schreibt.
4. Mikrocontroller nach einem der Ansprüche 1 bis 3, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit (160) in einer der Betriebsarten die Zentralprozessoreinheit (120) anhält und die Daten und die Adresse direkt aus dem nichtflüchtigen Speicher (130) liest und/oder in diesen schreibt.
5. Mikrocontroller nach einem der Ansprüche 1 bis 4, weiter dadurch gekennzeichnet, dass der Befehl für den nichtflüchtigen Speicher (130) einen internen Vorgang für denselben anzeigt, insbesondere einen Lesevorgang, einen Schreibvorgang oder einen Löschvorgang.
6. Mikrocontroller nach einem der Ansprüche 1 bis 5, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit (160) den Befehl für den nichtflüchtigen Speicher und die Adresse im Lesebetrieb oder im Löschbetrieb empfängt.

7. Mikrocontroller nach einem der Ansprüche 1 bis 6, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit (160) im Schreibbetrieb einen Satz aus dem Befehl für den nichtflüchtigen Speicher, einer Adresse und Daten empfängt.
8. Mikrocontroller nach einem der Ansprüche 1 bis 7, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit während eines Programmiermodus den internen Betrieb des nichtflüchtigen Speichers steuert, wobei die Zentralprozessoreinheit angehalten wird und in einem inaktiven Zustand bleibt.
9. Mikrocontroller nach Anspruch 8, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit den Befehl für den nichtflüchtigen Speicher sowie die Adresse und/oder die Daten über die serielle Schnittstelle von außerhalb des Mikrocontrollers seriell empfängt.
10. Mikrocontroller nach Anspruch 8 oder 9, weiter dadurch gekennzeichnet, dass die empfangenen Daten aus einem Steuerprogrammsatz bestehen, der eine Liste von Programminstruktionen und Programmdaten zum Programmieren des Mikrocontrollers umfasst.
11. Mikrocontroller nach einem der Ansprüche 8 bis 10, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit einen Steuerprogrammsatz und Daten seriell empfängt und den Steuerprogrammsatz und die Daten parallel in den nichtflüchtigen Speicher (130) schreibt.
12. Mikrocontroller nach einem der Ansprüche 1 bis 11, weiter dadurch gekennzeichnet, dass die Zentralprozessoreinheit einen Befehl für einen internen Vorgang des nichtflüchtigen Speichers über die Kommunikationsschnittstelle empfängt und decodiert, um mehrere Steuerbits der Speichersteuereinheit in Reaktion auf den decodierten Befehl zu schreiben.
13. Mikrocontroller nach Anspruch 12, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit die Zentralprozessoreinheit in Reaktion auf den ihr von dort zugeführten Schreibbefehl anhält, während ein Aktualisierungsprogramm in den nichtflüchtigen Speicher geschrieben wird.
14. Mikrocontroller nach Anspruch 12 oder 13, weiter dadurch gekennzeichnet, dass die in der Speichersteuereinheit enthaltene serielle Schnittstelle in einem inaktiven Zustand bleibt.
15. Mikrocontroller nach einem der Ansprüche 12 bis 14, weiter gekennzeichnet durch einen mit den Bussen verbundenen Pufferspeicher, eine in der Zentralprozessoreinheit enthaltene Registerdatei und einen im nichtflüchtigen Speicher enthaltenen Datenpuffer zum temporären Speichern eines Steuerprogrammsatzes und von Daten.
16. Mikrocontroller nach einem der Ansprüche 1 bis 15, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit einen Satz aus einem Steuerregister, einem Adressenregister und einem Datenregister beinhaltet, die funktionell mit einem der Busse oder der seriellen Schnittstelle verbunden sind.
17. Mikrocontroller nach Anspruch 16, weiter dadurch gekennzeichnet, dass das Steuerregister, das Adressenregister und das Datenregister über einen Adressdecoder an einen Adressbus gekoppelt sind.
18. Mikrocontroller nach Anspruch 16 oder 17, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit des weiteren einen Multiplexer zum selektiven Ausgeben eines Ausgangssignals eines Adressbusses oder des Adressenregisters in Reaktion auf ein Adressenauswahlsignal vom Steuerregister aufweist.
19. Mikrocontroller nach einem der Ansprüche 16 bis

18, weiter dadurch gekennzeichnet, dass die Speichersteuereinheit ein Sektorschutzhinweisregister aufweist, um ein falsches Schreiben in einen schreibgesperrten Zellensektor zu verhindern.

20. Mikrocontroller nach einem der Ansprüche 1 bis 19, weiter gekennzeichnet durch eine Programmier-/Lösch-Zeitsteuereinheit, um den nichtflüchtigen Speicher unabhängig von einer Systemumgebung, in welcher der Mikrocontroller angebracht ist, auf einer konstanten Programmier-/Löschgeschwindigkeit zu halten.

21. Mikrocontroller nach Anspruch 20, weiter dadurch gekennzeichnet, dass die Programmier-/Lösch-Zeitsteuereinheit einen Taktgenerator zur Erzeugung eines synchronen Programmier-/Lösch-Taktsignals unabhängig von einem an den Mikrocontroller angelegten Haupttakt aufweist.

22. Mikrocontroller nach einem der Ansprüche 1 bis 21, weiter dadurch gekennzeichnet, dass die Zentralprozessoreinheit während eines Programmiermodus Adressen und Daten von der Kommunikationsschnittstelle zur Speichersteuereinheit bewegt und die Speichersteuereinheit einen Lese- oder Schreibvorgang für den nichtflüchtigen Speicher ohne Einwirkung der Zentralprozessoreinheit ausführt.

---

Hierzu 8 Seite(n) Zeichnungen

---

30

35

40

45

50

55

60

65

- Leerseite -



Fig. 1

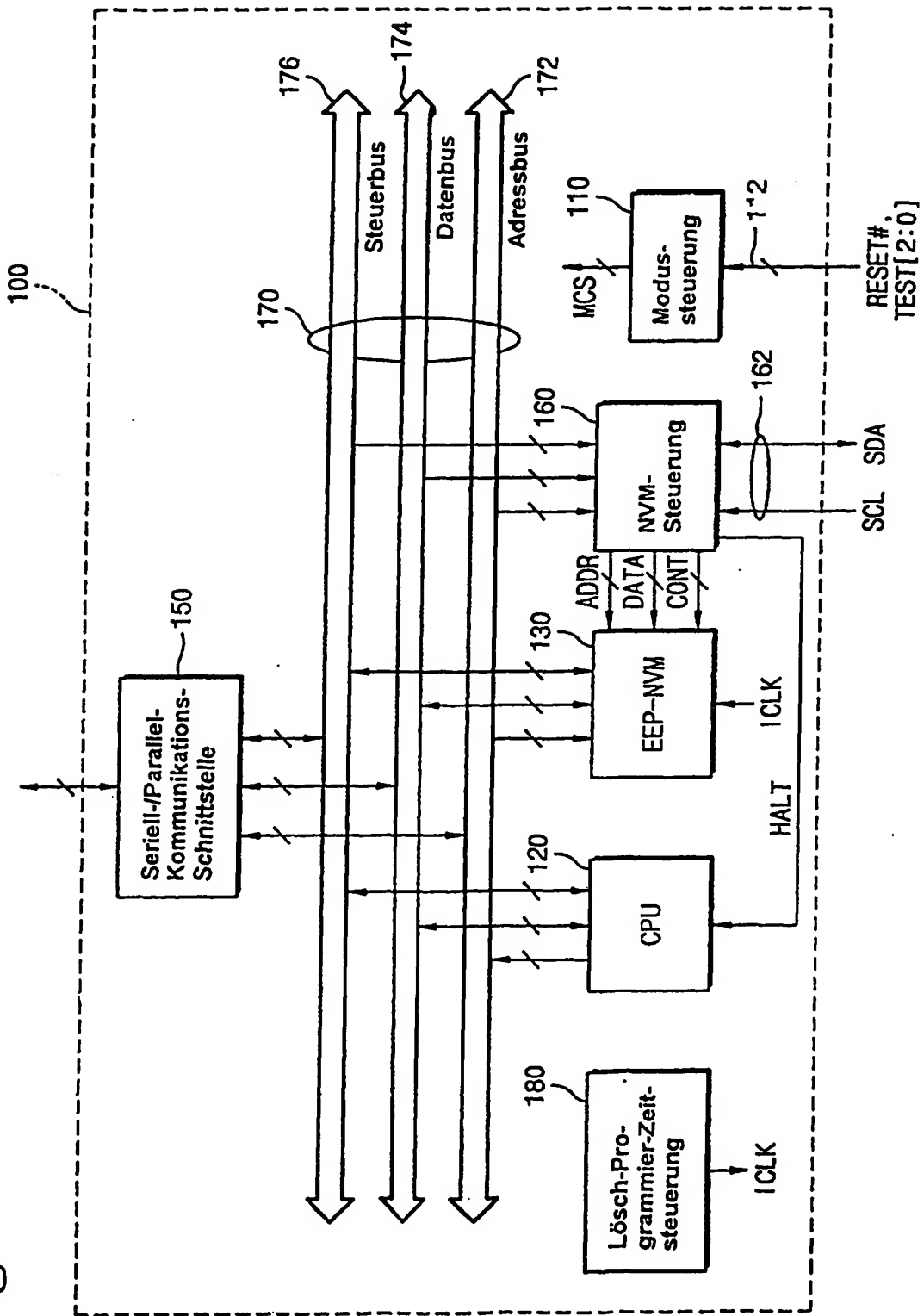


Fig. 2

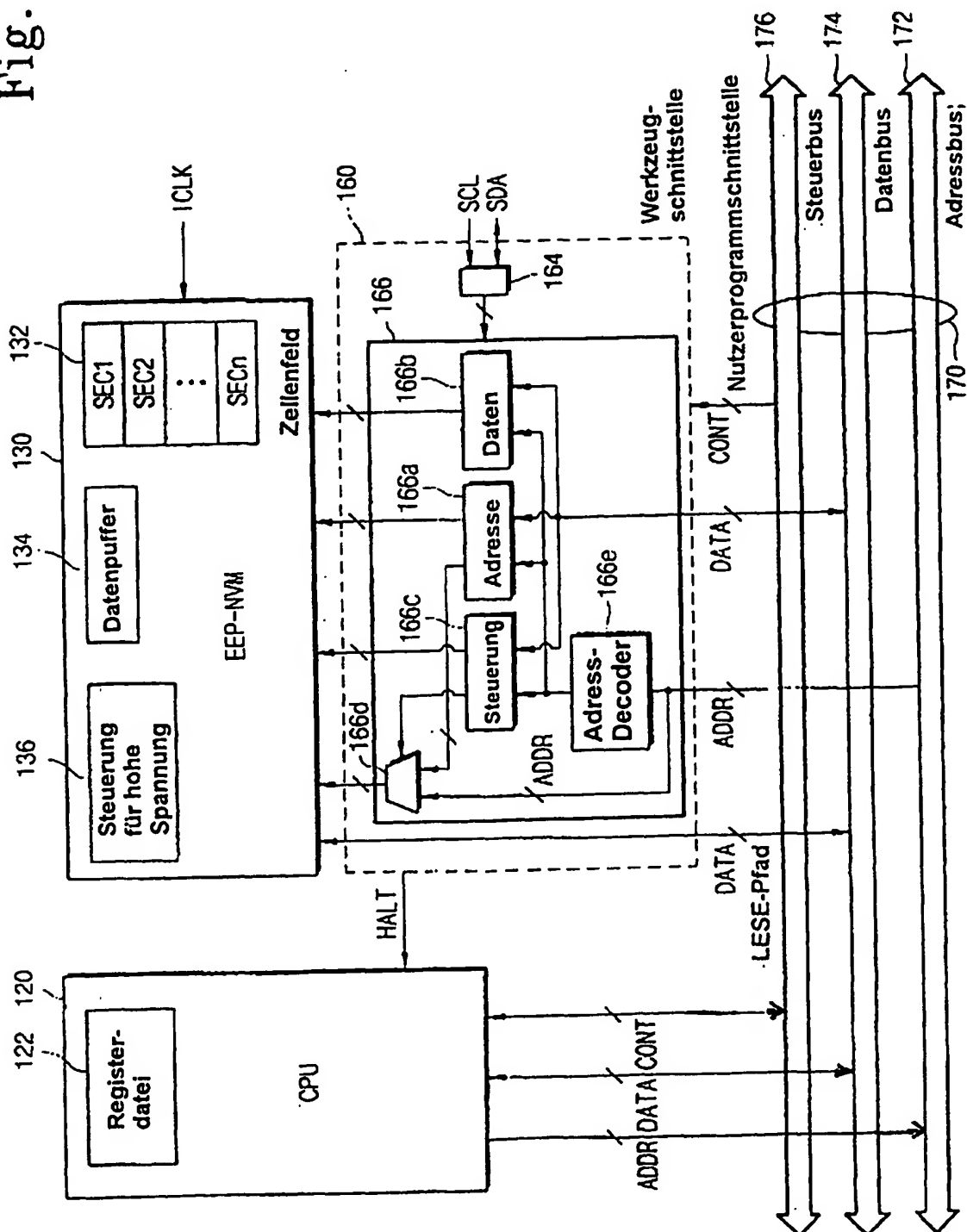
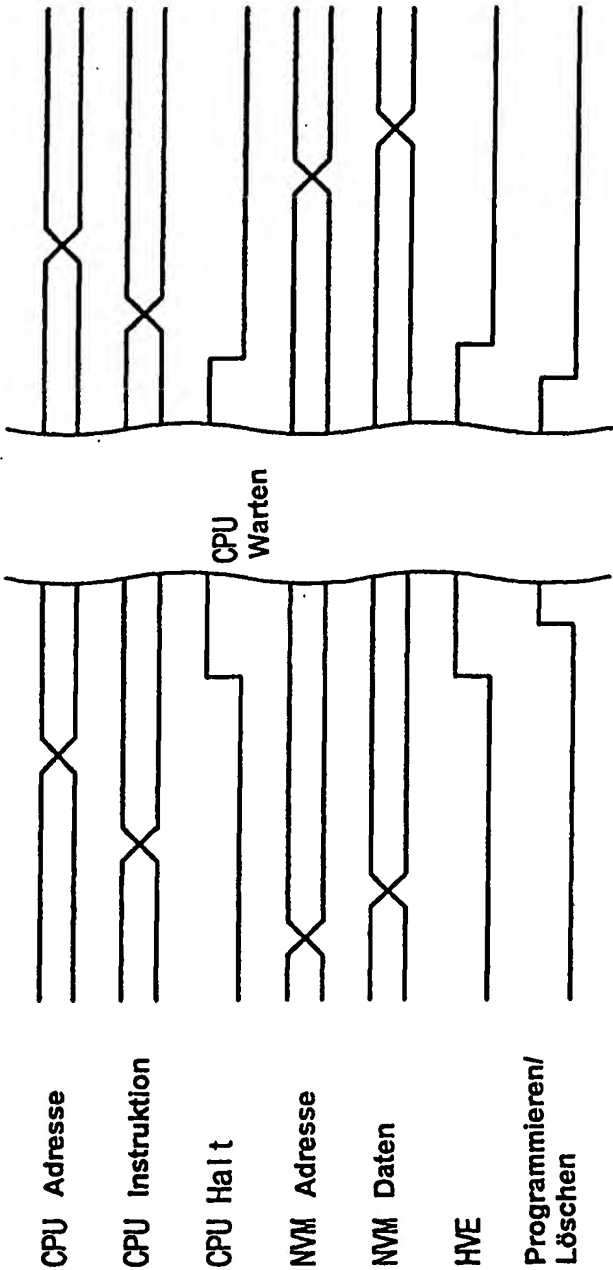


Fig. 3



**Fig. 4**

**Schlüsselregister.**

Verglichen mit einem 1. Wert

KR1							
-----	--	--	--	--	--	--	--

Verglichen mit einem 2. Wert

KR2							
-----	--	--	--	--	--	--	--

Verglichen mit einem 3. Wert

KR3							
-----	--	--	--	--	--	--	--

Verglichen mit einem 4. Wert

KR4							
-----	--	--	--	--	--	--	--

Fi. 5

## Nutzerprogrammiererregister

		FAST	ACC2	ACC1	ACCO	SDPD		STRSTP	SPGM	QSPGM	SERS	OSERS
--	--	------	------	------	------	------	--	--------	------	-------	------	-------

**FAST** : Speicherverschachtelungsmodus für hohe Geschwindigkeit

## ACC[2:0] : Speicherzugriffszyklus

SDPD : Softwaredatenschutzsperre

STRSTP : Betriebsstart/-stop#

**SPGM : Sektorprogrammierungsfreigabe**

**OSPGM** : Optionssektor(Smart-Option/Schutz)-Programmierfreigabe

**SERS : Sektorlöschfreigabe**

OSERS : Optionssektor(Smart-Option/Schutz)-Löschfreigabe

# Fig. 6

## Nutzerprogrammierfehlerregister

CFGERR	RGMERR	VTGERR					
--------	--------	--------	--	--	--	--	--

CFGERR : Konfigurationsfehler

PGMERR : Programmierfehler

VTGERR : Spannungsfehler

Fig. 7

Sektorschutzhinweisregister

U7	U6	U5	U4	U3	U2	U1											
LF	LE	LD	LC	LB	LA	L9	L8	L7	L6	L5	L4	L3	L2	L1	L0		

- L0 : 00000h-007FFh 2Kbyte

L1 : 00800h-00fffh 2Kbyte

L2 : 01000h-017ffh 2Kbyte

L3 : 01800h-01fffh 2Kbyte

L4 : 02000h-027ffh 2Kbyte

L5 : 02800h-02fffh 2Kbyte

L6 : 03000h-037ffh 2Kbyte

L7 : 03800h-03fffh 2Kbyte

L8 : 04000h-047FFh 2Kbyte

L9 : 04800h-04fffh 2Kbyte

LA : 05000h-057ffh 2Kbyte

LB : 05800h-05fffh 2Kbyte

LC : 06000h-067ffh 2Kbyte

LD : 06800h-06fffh 2Kbyte

LE : 07000h-07fffh 2Kbyte

LF : 07800h-07fffh 2Kbyte
- U1 : 08000h-0ffffh 32Kbyte

U2 : 10000h-17fffh 32Kbyte

U3 : 18000h-1ffffh 32Kbyte

U4 : 20000h-27fffh 32Kbyte

U5 : 28000h-2ffffh 32Kbyte

U6 : 30000h-37fffh 32Kbyte

U7 : 38000h-3ffffh 32Kbyte

Fig. 8

